What is claimed is:

1. A computerized method comprising:

providing a first vector of addressing values;

providing a second vector of operand values;

storing a first sequence of values to a sequence of addressed locations within a constrained area of memory, wherein each location's address is based at least in part on a corresponding one of the addressing values;

reading back from the sequence of addressed locations values resulting from the storing of the first sequence to obtain a second sequence of values;

comparing the first sequence of values to the second sequence of values to generate a bit vector representing compares and miscompares;

compressing the second vector of operand values using the bit vector;

using the first vector of addressing values as masked by the bit vector, loading a third vector register with elements from memory;

performing an arithmetic-logical operation using values from the third vector register and the compressed second vector of operand values to generate a result vector; and

using the first vector of addressing values as masked by the bit vector, storing the result vector to memory.

2. The method of claim 1, wherein addresses of the elements in memory are calculated by adding each respective addressing value to a base address of an object in memory.

3. The method of claim 1, wherein the arithmetic-logical operation is an addition operation that produces at least one element of the result vector as a summation of an element of the loaded third vector register and a plurality of respective elements of the original second vector of operand values corresponding to elements of the first vector of addressing values that had identical values.

4. The method of claim 1, wherein address values for the sequence of addressed locations within the constrained area of memory are each calculated using a truncated portion of each respective addressing value of the first vector of addressing values.

5. The method of claim 4, wherein data values of the first sequence of values are each formed by concatenating a portion of each respective addressing value of the first vector of addressing values to a respective one of a sequence of numbers.

6. The method of claim 1, wherein the constrained area of memory includes 2^N locations, wherein address values for the sequence of addressed locations within the constrained area of memory are each calculated by adding a base address to an N-bit portion of each respective addressing value of the first vector of addressing values, and wherein data values of the first sequence of values are each formed by concatenating a portion of each respective addressing value of the first vector of addressing values to a respective one of a consecutive sequence of integer numbers.

7. The method of claim 1, wherein for the loading of the third vector register with elements from memory, elements are loaded from locations specified by addressing values corresponding to bits of the bit vector that indicated a compare and no elements are loaded from locations specified by addressing values corresponding to bits of the bit vector that indicated a miscompare.

8. The method of claim 1, wherein the operations recited therein are executed in the order recited therein.

9. The method of claim 1, further comprising:

performing a first synchronization operation that ensures that the comparing the first sequence of values to the second sequence of values to generate the bit vector representing compares and miscompares effectively completes before the loading of the third vector register with elements from memory; and

performing a first synchronization operation that ensures that the storing the result vector to memory completes before subsequent passes through a loop.

10. A computer-readable medium having instructions stored thereon for causing a suitably programmed information-processing system to execute a method comprising:

providing a first vector of addressing values;

providing a second vector of operand values;

storing a first sequence of values to a sequence of addressed locations within a constrained area of memory, wherein each location's address is based at least in part on a corresponding one of the addressing values;

reading back from the sequence of addressed locations values resulting from the storing of the first sequence to obtain a second sequence of values;

comparing the first sequence of values to the second sequence of values to generate a bit vector representing compares and miscompares;

compressing the second vector of operand values using the bit vector;

using the first vector of addressing values as masked by the bit vector, loading a third vector register with elements from memory;

performing an arithmetic-logical operation using values from the third vector register and the compressed second vector of operand values to generate a result vector; and

using the first vector of addressing values as masked by the bit vector, storing the result vector to memory.

11. A computerized method comprising:

loading a first vector register with addressing values;

loading a second vector register with operand values;

storing a first sequence of values to a sequence of addressed locations within a constrained area of memory, wherein each one of these location's addresses in the constrained area of memory is based at least in part on a subset of bits of a corresponding one of the addressing values;

reading back from the sequence of addressed locations values resulting from the storing of the first sequence to obtain a second sequence of values;

comparing the first sequence of values to the second sequence of values;

selectively combining, with an arithmetic-logical operation, certain elements of the second vector of operand values based on results of the comparing;

using at least some of the first vector register of addressing values, loading a third vector register with elements from memory;

performing the arithmetic-logical operation using values from the third vector register and the combined second vector of operand values to generate a result vector; and

using the at least some of the first vector register of addressing values, storing the result vector to memory.

12. The method of claim 11, wherein addresses of the elements from memory are calculated by adding each respective addressing value to a base address.

13. The method of claim 11, wherein addresses of the elements from memory are calculated by performing a signed-addition operation of each respective addressing value to a base address of an object in memory.

14. The method of claim 11, wherein the arithmetic-logical operation is an addition operation that produces at least one element of the result vector as a summation of an element of the loaded third vector register and a plurality of respective elements of the original second vector of operand values corresponding to elements of the first vector register of addressing values having identical values.

15. The method of claim 11, wherein address values for the sequence of addressed locations within the constrained area of memory are each calculated using a truncated portion of each respective addressing value of the first vector register of

addressing values.

16. The method of claim 15, wherein data values of the first sequence of values are each formed by concatenating a portion of each respective addressing value of the first vector register of addressing values to a respective one of a sequence of numbers.

17. The method of claim 11, wherein the constrained area contains 2^N consecutive addresses, wherein address values for the sequence of addressed locations within the constrained area of memory are each calculated using an N-bit value derived from each respective addressing value of the first vector register of addressing values, and wherein data values of the first sequence of values are each formed by concatenating a portion of each respective addressing value of the first vector register of addressing values to a respective one of a consecutive sequence of integer numbers.

18. The method of claim 11, wherein for the loading of the third vector register with elements from memory, elements are loaded from locations specified by addressing values corresponding to indications that indicated compares and no elements are loaded from locations specified by addressing values corresponding to indications that indicated miscompares.

19. A computer-readable medium having instructions stored thereon for causing a suitably programmed information-processing system to execute the method of claim 11.

20. The method of claim 11,

wherein the constrained area contains 2^N consecutive addresses,

wherein address values for the sequence of addressed locations within the constrained area of memory are each calculated using an N-bit value derived from each respective addressing value of the first vector register of addressing values, wherein data values of the first sequence of values are each formed by combining at least a portion of each respective addressing value of the first vector register of addressing values to a respective one of a consecutive sequence of integer numbers,

wherein for the loading of the third vector register with elements from memory, elements are loaded from locations specified by addressing values corresponding to indications that indicated compares and no elements are loaded from locations specified by addressing values corresponding to indications that indicated miscompares,

wherein addresses of the elements from memory are calculated by adding each respective addressing value to a base address,

wherein the arithmetic-logical operation is a floating-point addition operation that produces at least one element of the result vector as an ordered-operation floating point summation of an element of the loaded third vector register and a plurality of respective elements of the original second vector of operand values corresponding to elements of the first vector register of addressing values having identical values, and

wherein for the storing of the result vector of elements to memory, elements are stored to locations specified by addressing values corresponding to indications that indicated compares and no elements are stored to locations specified by addressing values corresponding to indications that indicated miscompares.

21. A system comprising:

a first vector processor having:

a first vector register having addressing values;

a second vector register having operand values;

a third vector register;

a bit vector register;

circuitry that selectively stores a first sequence of values to a sequence of addressed locations within a constrained area of memory, wherein each location's address is based at least in part on a corresponding one of the addressing values;

circuitry that selectively loads, from the sequence of addressed locations, values resulting from the stores of the first sequence to obtain a second sequence of values;

circuitry that selectively compares the first sequence of values to the second sequence of values to generate bit values into the bit vector register representing compares and miscompares;

circuitry that selectively compresses the second vector of operand values using the values in the bit vector register;

circuitry that selectively loads the third vector register with elements from memory addresses generated from the first vector register of addressing values as masked by the bit vector register;

circuitry that selectively performs an arithmetic-logical operation on corresponding values from the third vector register and the compressed second vector of operand values to generate values of a result vector; and;

circuitry that selectively stores the result vector to memory.

22. The system of claim 21, further comprising

circuitry to calculate addresses of the elements in memory by adding each respective addressing value to a base address value.

23. The system of claim 21, wherein the arithmetic-logical operation is an addition operation that produces at least one element of the result vector as a summation of an element of the loaded third vector register and a plurality of respective elements of the original second vector of operand values corresponding to elements of the first vector register of addressing values that had identical values.

24. The system of claim 21, further comprising

circuitry to calculate address values for the sequence of addressed locations within the constrained area of memory using a truncated portion of each respective addressing value of the first vector register of addressing values.

25. The system of claim 24, further comprising

circuitry to generate data values of the first sequence of values by joining a portion of each respective addressing value of the first vector register of addressing values to a respective one of a sequence of numbers.

26. The system of claim 21, further comprising

circuitry to generate address values of the sequence of addressed locations within the constrained area of memory by adding a base address to an N-bit portion of each respective addressing value of the first vector register of addressing values; and

circuitry to generate data values of the first sequence of values by combining a portion of each respective addressing value of the first vector register of addressing values with a respective one of a consecutive sequence of integer numbers.

27. The system of claim 21, wherein the circuitry that selectively loads the third vector register with elements from memory only loads element from locations specified by addressing values corresponding to bits of the bit vector that indicated a compare.

28. The system of claim 21, further comprising:

synchronization circuitry that ensures that the comparing the first sequence of values to the second sequence of values to generate the bit vector representing compares and miscompares effectively completes before the loading of the third vector register with elements from memory, and that ensures that the storing the result vector to memory completes before subsequent passes through a loop.

29. The system of claim 21, further comprising:

a second vector processor having:

a first vector register having addressing values;

a second vector register having operand values;

a third vector register;

a bit vector register;

circuitry that selectively stores a first sequence of values to a sequence of addressed locations within a constrained area of memory, wherein each location's address is based at least in part on a corresponding one of the addressing values;

circuitry that selectively loads, from the sequence of addressed locations, values resulting from the stores of the first sequence to obtain a second sequence of values;

circuitry that selectively compares the first sequence of values to the second sequence of values to generate bit values into the bit vector register representing compares and miscompares;

circuitry that selectively compresses the second vector of operand values using the values in the bit vector register;

circuitry that selectively loads the third vector register with elements from memory addresses generated from the first vector register of addressing values as masked by the bit vector register;

circuitry that selectively performs an arithmetic-logical operation on corresponding values from the third vector register and the compressed second vector of operand values to generate values of a result vector; and;

circuitry that selectively stores the result vector to memory; and

synchronization circuitry that ensures that the comparing the first sequence of values to the second sequence of values to generate the bit vector representing compares and miscompares effectively completes in both the first and second vector processors before the loading of the third vector register with elements from memory in either processor, and that ensures that the storing the result vector to memory completes before subsequent passes through a loop.

30. A system comprising:

a first vector register;

a second vector register;

a third vector register;

a bit vector register;

means for loading the first vector register with addressing values;

means for loading the second vector register with operand values;

means for storing a first sequence of values to a sequence of addressed locations within a constrained area of memory, wherein each one of these location's addresses in the constrained area of memory is based at least in part on a subset of bits of a corresponding one of the addressing values;

means for loading from the sequence of addressed locations values resulting from the storing of the first sequence to obtain a second sequence of values;

means for comparing the first sequence of values to the second sequence of values;

means for selectively combining, with an arithmetic-logical operation, certain elements of the second vector of operand values based on results of the comparing;

means for loading a third vector register with elements from memory address locations generated using at least some of the first vector register of addressing values;

means for performing the arithmetic-logical operation using values from the third vector register and the combined second vector of operand values to generate a result vector; and

means for storing the result vector to memory.

31. A system comprising:

a first vector register that can be loaded with addressing values;

a second vector register that can be loaded with operand values;

a third vector register that can be loaded with operand values from memory locations indirectly addressed using the addressing values from the first vector register;

a circuit that determines element addresses of the first vector register that have a value that duplicates a value in another element address;

a circuit that selectively adds certain elements of the second vector of operand values based on the element addresses the duplicated values;

a circuit that uses indirect addressing to selectively load the third vector register with elements from memory;

a circuit that selectively adds values from the third vector register and the second vector of operand values to generate a result vector; and

a circuit that selectively stores the result vector to memory using indirect addressing.

32. The system of claim 31, further comprising:

an adder that generates addresses of the elements from memory by adding each respective addressing value to a base address.

33. The system of claim 31, further comprising:

an adder that generates addresses of the elements from memory by a signed-addition operation of each respective addressing value to a base address of an object in memory.

34. The system of claim 31, wherein the circuit that selectively adds certain elements performs one or more addition operations using those values from a plurality of respective elements of the original second vector of operand values corresponding to elements of the first vector register of addressing values having identical values.